SIMPLE
Internet-Draft
Expires: August 9, 2006

E. Burger
Brooktrout Technology, Inc.

February 5, 2006

Instant Message Delivery Notification (IMDN) for Common Presence and Instant Messaging (CPIM) draft-burger-simple-imdn-03

Status of this Memo

Copyright Notice

Abstract

This document describes a mechanism for instant message delivery notification (IMDN) in the CPIM (Common Presence and Instant Messaging) environment. The mechanism follows the procedures of ESMTP message delivery notification (MDN).

Table of Contents

1. Document Conventions

This document refers generically to the sender of a message in the masculine (he/him/his) and the recipient of the message in the feminine (she/her/hers). This convention is purely for convenience and makes no assumption about the gender of a message sender or recipient.

In this document, the term "CPIM header" refers to the message metadata headers encapsulated in a Message/CPIM object [1].

The term "IM" refers to Instant Message.

The term "Requesting UAC" is the User Agent Client that sends the message the user would like a disposition notification for.

The term "Reporting UAS" is the User Agent Server that sends the disposition notification back to the Requesting UAC.

The term "B2BUA" refers to a Back-to-Back User Agent. IM B2BUA's implement gateways and list exploders, amongst other functions.

If you missed it in the title, the term "IMDN" is an Instant Message Delivery Notification. The IMDN indicates the disposition of the message after the message is available to the recipient.

> NOTE: Text like this, offset from the margin and starting with the word "NOTE:", is not normative and present only for discussion or explanation purposes.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [2].

This document uses the augmented Backus-Naur Form (BNF) as described in RFC4234 [3] for all syntax specification uses, other than XML.

Examples and discussion of CPIM headers, for clarity, do not include the leading name space identifier.

2. Introduction

In many user-to-user message exchange systems, message senders often wish to know if the human recipient actually read a message. Most messaging protocols, including CPIM sessions [4], ensure reliable delivery of a message to the recipient. However, they cannot report when a human user actually reads the message.

Electronic Mail [12] deals with this situation with message delivery notifications [5]. After the recipient views the message, her mail user agent generates a message delivery notification, or MDN. The MDN is an e-mail that follows the format prescribed by RFC2298 [5]. The fixed format ensures that an automaton can process the message. Even though a MDN is a normal e-mail message, a MDN cannot request a receipt, in order to prevent notification loops.

The mechanism described here uses a CPIM header to indicate an IMDN request. By using a CPIM header, we abstract the request outside the transport protocol. This enhances interoperability between different IM systems because the request is at the message level, not transport level. Likewise, the mechanism does not rely on session-mode versus pager-mode or SIP transport or any particular SIP or other response codes.

Since the security and privacy considerations have a tremendous influence on a number of design decisions that may at first seem counter-intuitive, the Privacy Considerations (Section 4) and Security Considerations (Section 3) sections appear in the front of this document.

The basic protocol flow is as follows. A message sender marks a message for disposition notification. At a certain point in time, the recipient's instant message user agent determines the recipient has read the message or otherwise disposed the message. The mechanism by which an instant message user agent determines its user has read a message is beyond the scope of this document. At that point, the recipient's instant message user agent automatically generates a notification message to the sender. This notification message is the Instant Message Disposition Notification (IMDN).

Note there are numerous circumstances under which the instant message user agent may not send a notification. The following sections describe some of these situations.

3. Security Considerations

All of the security issues raised in RFC2298 [5] apply here. For review, they are forgery and denial of service attacks, confidentiality, and non-repudiation. Note that signing CPIM messages helps in this respect.

The threats in the IMDN system, over and beyond the threats inherent to CPIM [4] include the following:

- A malicious endpoint attempts to send messages to a user that would normally not wish to receive messages from that endpoint by convincing the IMDN system to "bounce" an IMDN from an unsuspecting endpoint to the user.
- A malicious endpoint attempts to flood a user with IMDNs by convincing a B2BUA list exploder to send IMDN responses to an unsuspecting user.
- A malicious node in the network that attempts to modify an IMDN from a Reporting UAS.
- A malicious B2BUA attempts to forward an IMDN from a Reporting UAS to the Recipient UAC, where the Reporting UAS would not normally forward the IMDN to that Recipient UAC if the identity of the Reporting UAS knew the identity of the Recipient UAC.
- A malicious endpoint that attempts to fish for a Request-URI of a UAS beyond a B2BUA, where the UAS would normally wish to keep its identity private from the malicious endpoint.
- A malicious node in the network that attempts to eavesdrop on IMDN traffic to, for example, learn Request-URI or traffic pattern information.
- A malicious node in the network attempts to stage a denial of service attack on a B2BUA by requesting a large list expansion with a request for consolidated IMDN processing.

Attacks the protocol cannot protect against include the following:

- A malicious B2BUA directly revealing the identity of a downstream UAS that would not normally wish its identity revealed to such a UAS. Keeping such information private is a B2BUA implementation issue.
- A malicious Reporting UAS that alters the time of the report. There is no protocol mechanism for ensuring the UAS does not lie about the time or purposely holds an IMDN for transmission to make it appear the user disposed of a message later than they actually did.
- A deletion attack on an IMDN. This is a trade-off between privacy and security. The privacy considerations allow the Reporting UAS to silently ignore an IMDN request. Any mechanism that would reliably indicate that a malicious node deleted a Reporting UAS' message would also serve the purpose of detecting a Reporting UAS that chose not to issue an IMDN.

To combat eavesdropping, modification, and man-in-the-middle attacks, we require some level of authentication and integrity protections. That said, there are circumstances where strong integrity would be overkill. The presumption is the sender of the IM has and sets the expectation for the level of protection. The procedures for integrity protection are as follows.

- o If the Reporting UAS has a certificate, the Reporting UAS MUST sign the IMDN.
- o If the IM is encrypted, the Reporting UAS MUST encrypt the IMDN body, as an attacker may attempt to discern the user's activity profile and identity from sniffing IMDNs on the network.
- o The two above rules are cumulative.

Reporting UAS' MUST be capable of loading a user certificate.

Replay and message insertion attacks are unlikely in an IMDN environment, as the Message-ID cannot be identical within a given session, and the Requesting UAC has the ability to maintain the state of Message-ID's sent for later correlation. Moreover, the instant message itself MUST have the Message-ID sent securely to remove the possibility of an eavesdropper learning the Message-ID.

To combat surreptitious delivery of messages by embedding them in IMDN's, as is done today by spammers using MDN's, an IMDN MUST NOT include a copy of the original message.

An attacker can mount a distributed denial of service attack on a node by sending lots of messages to the node with IMDN requests. Note that this is the same problem as there is without IMDN; IMDN simply linearly increases the load on the node under attack. One can mitigate, but not eliminate this threat by the UAS immediately ignoring requests that are not authenticated.

Likewise, an attacker can mount a denial of service attack on a B2BUA by asking the B2BUA to explode a large list and to direct the B2BUA to consolidate the IMDN's from the targets of the list.

4. Privacy Considerations

As suggested by RFC2298 [5], it is strongly RECOMMENDED that the user agent obtain the user's consent before sending an IMDN. Circumstances where the user agend does not ask for the user's consent include instant messaging systems that, for regulatory reasons, are required to issue an IMDN, such as in the healthcare field or financial community.

A user agent can obtain such consent by a prompt or dialog box on a per-message basis, globally through the user's setting of a preference, or other, user-configurable mechanism. The user might also indicate globally that IMDNs are never to be sent or that a "denied" IMDN is always sent in response to a request for an IMDN.

The protocol MUST enable the recipient of the IM to keep the message disposition private. That is, only the sender is able to read the

IMDN body text. The transmission of the IMDN SHOULD be end-to-end encrypted, if physically possible. If the Originating UAC encrypted the IM, the Reporting UAS MUST encrypt the IMDN and wrap the result with S/MIME [6].

5. State Sharing

One of the design questions for the IMDN design is where to store message state. This becomes a question when we introduce B2BUA's. If there were no B2BUA's, then the Reporting UAS simply sends the IMDN directly to the sender, and the sender retains the state of messages sent that are awaiting IMDN's. However, since we have B2BUA's, we have a choice. One option is for the Requesting UAC to record the state of messages sent and have stateless B2BUA's. Another option is for intermediary B2BUAs to record the state of messages sent and always forward IMDN's back to the immediately preceding message requestor.

The trade-offs are as follows:

- End-to-End State Sharing
  - Pro: The actual recipient sends the IMDN directly to the actual sender. It is quite likely the path may be different. For example, while the request may traverse a list exploder (B2BUA), the IMDN's will go directly to the sender.
  - Pro: With the Reporting UAS sending the report directly to the Requesting UAC, it is possible to keep the disposition private with respect to intermediaries.
  - Pro: Only the endpoints share state. Network failures do not impact IMDN delivery. Users should know when their user agent fails and can act accordingly.
  - Pro: No matter what, the Requesting UAC should store state for messages is sends and has an interest in correlating IMDN's. There is no additional burden to the network or user agent.
  - Pro: The Reporting UAS knows the direct recipient of the IMDN, so it can use more sophisticated algorithms to decide if or what kind of IMDN to generate. Of course, a B2BUA can always hide the true recipient of an IMDN by requesting the report on its own behalf, as it is a full UAC. However, the Reporting UAS can chose not to release information to untrusted B2BUA's.
  - Con: Slightly more complex protocol, and requires authenticated hop-by-hop transport to combat spam and man-in-the-middle attacks.
  - Con: Devices with limited resources and a high likelihood of total failure, such as a mobile phone, will lose their IMDN request state on total failure.

- Intermediary State Sharing
  - Pro: Supports endpoints with limited resources or a high likelihood of total failure, so long as all messages go through a B2BUA that records state.
  - Pro: Simpler protocol: IMDN's always go to the "last" user agent up the relay chain.
  - Pro: The B2BUA hides a recipient's IMDN reply to address, yet still let the Reporting UAS issue an IMDN the B2BUA will forward to the Requesting UAC.
    NOTE: Is this important? It is saying that I am sending an IM to someone who I do not want to let talk back to me. That does not really fit the model of IM, does it?
  - Con: Asking the B2BUA to take the role of IMDN forwarder means the Requesting UAC loses any chance of getting end-to-end IMDN's.
  - Con: B2BUA's will have a tremendous amount of state to store, especially given their location in the network.

The consensus of the work group is to use intermediary state sharing as it results in a simpler protocol. The protocol does leave the potential for future end-to-end state sharing by allowing a token to be the value of the Disposition-Notification header. The most likely value for this token is the URI to send the IMDN directly to.

6. Overview

6.1. Data Elements

The data elements required for IMDN include elements that help correlate notifications with messages, indicate whether or not to generate a notification message, and, of course, the disposition of the message itself.

The following list enumerates the data elements of the IMDN in detail.

- A protocol data element that indicates an IMDN request
- The Original Message Identifier uniquely identifies the original message. Currently there is no unique message identifier in CPIM. Thus we will define one in this document.
- The Reporting UAS identifies the UAS generating the IMDN. The reporting UAS might not be the "sender" of the IMDN, as there may be relays [13] between the Reporting UAS and the requesting UAC.
- The Original Recipient URI identifies the original URI the requesting UAC sent the message to. This may not be the same as the Reporting UAS, as message delivery to the original URI may have resulted in a list expansion. Section 6.3 describes B2BUA procedures in detail.

- o An indicator to keep the Original Recipient URI private.
- o The Message Disposition identifies the eventual disposition of the message, such as read, deleted, and so on.
- o The Disposition Time indicates when the disposition time at the Reporting UAS.

The Requesting UAC needs to indicate to the Reporting UAS to generate an IMDN. The Requesting UAC can indicate whether list exploders or gateways should report on their receipt of the message or report on the actual end recipient's receipt of the message.

6.2. Disposition States

There are three broad categories of disposition states. They are read, processed, error, and denied.

6.2.1. read

The "read" state is straightforward. The read state indicates the Recipient's UAS displayed the message to the user.

Since there is no positive acknowledgement from the user, one cannot determine a priori the user actually read the message. Thus one MUST NOT use the protocol described here as a non-repudiation service.

6.2.2. processed

The target URI of the message is a B2BUA. The B2BUA's UAS indicates it successfully received and expanded or relayed the message. However, there MUST NOT be further notifications for this message (see Section 6.3. See Section 8.1.2 for how the Requesting UAS can drive the generation of the processed notification when List-Action is first-recipient.

6.2.3. error

The error state indicates there was some processing error that makes it impossible or unlikely for the user to get the message.

6.2.4. denied

The denied state indicates the target URI does not allow notifications. This could be for any reason, including a general policy to not send notifications, denying notifications to the particular sender, or by user direction on a per-message basis. For privacy reasons, the UAS MUST NOT give the reason for denial.

6.3. B2BUAs

The IMDN framework presented here supports back-to-back user agents (B2BUAs) that forward message requests. This models most approaches for list expansion, including SIP URI lists [16]. It also models most gateway mechanisms.

When a user sends a message to a B2BUA URI, there are two options for interpreting "delivery". One option is to consider the message delivered to the list exploder URI itself. This is a strict interpretation of "delivery", as the list exploder URI resolves to the B2BUA UAS. What happens on the other side of the list exploder, namely, the re-origination of a bunch of messages, nominally related to the first message, has no relation in a protocol sense to the original message.

The other option is to consider the message delivered to the ultimate recipients of the list. On the one hand, this is what users expect, especially if the list is emulating a chat room. On the other hand, this could result in a storm of responses, which the user does not want.

If the B2BUA will be forwarding an IMDN from a downstream endpoint, it will encapsulate the IMDN. This enables signatures over the original message. Moreover, since the end system has the Original From URI, it has the potential to encrypt the IMDN using, for example, S/MIME [6], for the original sender, resulting in end-to-end security.

Gateway processing is identical to list exploder processing, in that this mechanism considers a gateway to be a list exploder with a single destination.

To ease interpretation of the IMDN at the B2BUA and original Requesting UAC, the B2BUA MUST preserve the original URI the Requesting UAC sent the message to. That is, it must carry the value of the To header in the Original-To CPIM header.

7. Namespace

Per CPIM [1], IMDN uses a namespace for the CPIM headers. The namespace is
urn:ietf:params:cpim-headers:imdn

All of the header definitions in this document refer to this namespace.

NOTE: If one does not specify the name space in one's CPIM message, YOU WILL NOT GET THE BEHAVIOR DESCRIBED IN THIS DOCUMENT.

8. Requesting UAC Behavior

8.1. IMDN Request Generation

To request the generation of an IMDN, the Requesting UAC MUST include the Disposition-Notification and Message-ID headers. The Requesting UAC MAY also include a List-Action header to provide down-stream B2BUA's with the user's desire for IMDN reporting by the final target of B2BUA expansion or the B2BUA itself. B2BUA's SHOULD include the Original-From header, with the value of the inbound From header, unless privacy considerations require the B2BUA to not transmit the Original-From header. Likewise, B2BUA's SHOULD copy the value of the inbound Message-ID into the outbound Original-Message-Id header.

If the Requesting UAC insists on the possibility of an IMDN being generated, the UAC MUST include the "Require: imdn.Message-Disposition-To" header, where "imdn" is a reference to the name space (the "NS" header). While this ensures the Reporting UAS is capable of generating an IMDN, there is no guarantee that it actually will generate an IMDN. See the Privacy Considerations (Section 4) section for more discussion on this point.

8.1.1. Disposition-Notification

To mark a message for disposition notification, the sender MUST include a Disposition-Notification CPIM header in the CPIM part of the request.

If the sender requires a notification, the message MUST include a CPIM Require header requiring the processing of the Disposition-Notification CPIM header. Note that if the Recipient UAS does not support IMDN, then the UAS will reject the message. In addition, the Recipient UAS SHOULD NOT display the message.

If the sender does not require Disposition-Notification, and the recipient's instant message user agent does not support IMDN, then even though the recipient may read the message, the sender will not receive a notification report.

Note that the choice of including a Require header is entirely a local matter to the sender. Some instant messaging user agents may make this a per-receipt request option. Another opinion is the Requesting UAC should never use the Require header to improve interoperability with non-IMDN clients. However, in that case the

sender will not know if his message had no report because the recipient did not read it or if the recipient's UAS was simply unaware of IMDN. Thus the decision to use the Require header is entirely outside the scope of this document.

The syntax of the Disposition-Notification CPIM header is
mdn-request-header = "Disposition-Notification" ":" NULL / token-list

token-list = token-list token / token

For CPIM conferences, a message with a Disposition-Notification header will result in all recipients performing IMDN processing. If this is not desirable, the sending system MUST send multiple messages with the appropriate requests (IMDN or not).

Systems sending an IMDN MUST NOT include a Disposition-Notification header.

At this time, there are no Disposition-Notification parameters or tokens defined. Adding Disposition-Notification parameters MUST be by a Standards Track RFC.

8.1.2. List-Action

If the user sends a message to a B2BUA, such as a list expander or gateway, the Requesting UAC MAY include a List-Action header. The List-Action header indicates how the B2BUA should handle IMDN generation.

Values for List-Action are:

final-recipient	This is a request for the B2BUA to request IMDN's from the subsequent requests and relay the IMDN to the Requesting UAC.
first-recipient	This is a request for the B2BUA to generate an IMDN for the B2BUA's receipt of the message. That is, the disposition reflects the B2BUA's processing of the message, not any down-stream messages.
{other}	The Reporting UAS MUST treat unrecognized values for List-Action as "first-recipient". Definitions for new values MUST include optional CPIM REQUIRE tags to ensure interoperability.

If the Requesting UAC does not specify List-Action, the default List-Action is first-recipient.

The syntax of the List-Action header is as follows.

```
list-action-hdr = "List-Action" ":" SP list-actions
list-actions    = "final-recipient"
                / "first-recipient"
                / token
```

8.1.3. Original-From

A Requesting UAC MAY include its From identifier as the value to the Original-From header. If there is no Original-From header, the value of the From header is used. If there is no Original-From header in the message, a B2BUA MUST populate the Original-From value with the From identifier from the inbound message. If there is an Original-From header in the message, a B2BUA MUST pass the Original-From header to the recipient URI(s). This ensures that notifications from lists of lists will work, and that end-to-end encryption of IMDN's will work.

If a Requesting UAC wishes to keep her URI private through a B2BUA, then the Reporting UAC includes the Original-From header, but with a NULL value.

The syntax of the Original-From is as follows.

```
original-from-header  =  "Original-From" ":" SP from-whom

from-whom             =  "" / from-address

from-address          =  [ Formal-name ] "<" URI ">"
```

RFC3862 [1] section 4.1 defines "Formal-name".

8.1.4. Message-ID

A UAC MUST include a globally unique Message-ID. It is necessary for the Message-ID to be unique to the UAC in order for the UAC to be able to exactly correlate IMDN's with the messages they refer to. It will be necessary for the Message-ID to be globally unique in order to support frameworks such as message tracking [15] in the future. Since it is easy enough to make the Message-ID globally unique now, we mandate it here so that message tracking will be easier in the future.

A Reporting UAC MUST be prepared to handle a Message-ID token of at least 4095 octets.

The syntax of the Message-ID is as follows.

```
message-id-hdr = "Message-ID" ":" SP token
```

A B2BUA generates new messages, and thus the Message-ID will be new.

8.1.5. Original-Message-ID

Since a B2BUA generates new messages, and thus new Message-ID's, we need a mechanism for the Reporting UAS to insert the appropriate Message-ID in the IMDN. To do this, a B2BUA inserts an Original-Message-ID header with the value of the Message-ID. If there is already an Original-Message-ID header, then the B2BUA MUST preserve the value in the outbound request, unless the request forbids it. The request may forbid it if, for example, the List-Action is first-recipient. If there is no Original-Message-ID present in a message delivered to a B2BUA for subsequent forwarding, the B2BUA MUST copy the value of the Message-ID header of the inbound message to be the value of the Original-Message-ID header of the outbound message(s).

The syntax of the Original-Message-ID is as follows.
original-message-id-hdr = "Original-Message-ID" ":" SP token

8.2. IMDN Reception Processing

Once a Requesting UAC sends a message with an IMDN request, it SHOULD preserve the message context, principally the Message-ID, and other user-identifiable information such as the message subject or content. Without preservation of the message context, the Requesting UAC will not be able to correlate the IMDN with the outbound request. The Requesting UAC may find it impossible to preserve message state if it has limited resources or does not have non-volatile memory and then loses power.

How long to preserve the state is an implementation choice. However, the Requesting UAC SHOULD keep the state for at least 5 minutes, unless that is physically impossible due to the characteristics of the Requesting UAC.

It is RECOMMENDED that a Requesting UAC not notify the user if the Requesting UAC receives an IMDN that does not correlate to a message the Requesting UAC sent. This is to prevent IMDN spoofing. Clearly, if a requesting UAC loses its sent message state, the client may use a different display strategy in response to apparently unsolicited IMDN's.

A Requesting UAC MUST NOT issue an IMDN in response to an IMDN, even if that IMDN incorrectly includes a Disposition-Notification header.

9. Reporting UAS Operation

9.1. General Operation

When the Reporting UAS receives a CPIM message with a Disposition-Notification CPIM header, the Reporting UAS SHOULD generate an IMDN. Security Considerations, Privacy Considerations, and local policy all may prevent the generation of an IMDN.

The Reporting UAS MUST NOT send more than one IMDN in response to an IMDN request. That is, once an IMDN has been issued on behalf of a recipient, no further IMDNs may be issued on behalf of that recipient, even if another disposition is performed on the message. For example, if the user reads and then deletes the message, the UAS will send a single read notification. The delete operation in this case will not generate an additional IMDN. Likewise, a B2BUA receiving a List-Action of first-recipient MUST NOT relay IMDN's from down-stream UAS's to the original Requesting UAC, as the original Requesting UAC has asked only for an IMDN from the B2BUA.

A system receiving an instant message disposition notification MUST NOT generate a message disposition notification in response to that notification, even if the request includes a Disposition-Notification header.

A system sending an IMDN MUST NOT include the Disposition-Notification-To header.

A CPIM message that requests an IMDN but does not include the required Message-ID header is malformed and the UAS MUST reject the request using the appropriate protocol mechanism for rejecting a malformed request.

NOTE: We could be helpful here and create a new SIP result code for this situation. We can do that if needed.

The Reporting UAS MUST copy the incoming CPIM Subject: header as the IMDN CPIM Subject: header.

9.2. Recipient is the End User UAS

If the recipient of a CPIM message with a well-formed IMDN request is the end-user user agent server, then

- If the user read the message, then the UAS SHOULD generate a read IMDN, mindful of the privacy considerations enumerated in Section 4.
- If the UAS automatically deleted the message, or the user deleted the message without reading it, then the UAS SHOULD generate a processed IMDN, mindful of the privacy considerations enumerated in Section 4.

- If the UAS' policy is to deny IMDN to the requestor, or if the user requests a denial report to the UAC, the UAS SHOULD generate a denied IMDN, mindful of the privacy considerations enumerated in Section 4.
- If the UAS' policy is to ignore IMDN requests, or the user requests the supression of a given IMDN report, the UAS MUST silently ignore the IMDN request.

If the Reporting UAS has a certificate, it MUST sign the IMDN it generates using S/MIME [6]. The Reporting UAS MUST be capable of loading a certificate for signing IMDN's.

If the Requesting UAC encrypts the IM, the Reporting UAS MUST encrypt the IMDN. For assistance in this task, the URI of the endpoint requesting the IMDN is in the Original-From header.

The Reporting UAS MUST use the format and fill in the content of the IMDN as described in Section 10.

9.3. Recipient is a B2BUA

If the Recipient UAS is a back-to-back user agent (B2BUA), such as a list exploder or messaging gateway, then the action taken depends on the value of List-Action. If there is no List-Action header, or the UAS does not understand the value of the List-Action header, the UAS takes the "first-recipient" action.

9.3.1. first-recipient

If the List-Action is "first-recipient" or there is no List-Action specified, then the Recipient UAS issues an IMDN using the following procedures.

- If the B2BUA forwards the message, it SHOULD return an "processed" IMDN, mindful of the privacy considerations enumerated in Section 4.
- If there was an error processing the message, the B2BUA SHOULD return an "error" IMDN, mindful of the privacy considerations enumerated in Section 4.

Including a Disposition-Notification header in the forwarded messages is a matter of local policy. However, if the List-Action is first-recipient or unspecified, the B2BUA MUST NOT relay down-stream IMDN's to the original Requesting UAC.

The Reporting UAS MUST use the format and fill in the content of the IMDN as described in Section 10.

9.3.2. final-recipient

If the List-Action is "final-recipient", then the B2BUA SHOULD forward the Message-Disposition and List-Action headers to down-stream destinations. One condition where the B2BUA might not forward these headers is where the B2BUA knows the down-stream destination will not honor or is not capable of honoring the IMDN request. In the latter case, the B2BUA SHOULD return an "processed" IMDN. In the former case, local policy will decide whether to return a denied IMDN, processed IMDN, or not return an IMDN at all.

When the B2BUA receives an IMDN from the Reporting UAS, the B2BUA will encapsulate the IMDN from the downstream UAS and send the response to the UAC that generated the upstream request. The B2BUA MUST verify the Original-Message-ID header matches a Message-ID of a previous incoming request.

How long to keep the Message-ID state is a local matter. We RECOMMEND it be at least 5 minutes.

If the B2BUA receives an IMDN that does not match an existing Message-ID, the B2BUA MUST discard the IMDN.

9.3.3. No List-Action Specified

If there is no List-Action header, or there is a List-Action header with no value, the Reporting UAS MUST follow the procedures for first-recipient.

9.3.4. Unknown List-Action Specified

If there is a List-Action header, but the Reporting UAS does not recognize the value of the List-Action header, the Reporting UAS MUST follow the procedures for first-recipient.

10. IMDN Format

The IMDN is an XML [7] document. The document MUST be well formed and MUST be valid according to the schema presented in Section 12.2. The namespace identifier for elements defined by this specification is a URN [8], using the namespace identifier 'ietf' as defined by IETF URN Namespace [9] and extended by the IETF XML Registry [11]. The URN is
urn:ietf:params:xml:ns:imdn

The root element is <imdn>. The disposition tag takes the value described in Section 6.2.

10.1. disposition

The Reporting UAS MUST include the message disposition in the disposition tag.

We use a string value, rather than an attribute value, to enable future addition of state strings

The value of the disposition tag may take the following values:

read	The message was read.
processed	The message was processed.
error	There was an error processing the message.
denied	The Reporting UAS denies reporting the disposition of the message.

10.2. original-message-id

The Reporting UAS MUST include the value of the Message-ID of the IM as the value of the original-message-id tag.

10.3. original-recipient-uri

The Reporting UAS MUST include the value of the original message's Original-From as the value of the original-recipient-uri tag. If the Reporting UAS specified a NULL Original-From, then the Reporting UAS MUST return an empty original-recipient-uri value.

10.4. reporting-uas-uri

The Reporting UAS SHOULD include its URI in the reporting-uas-uri tag. One condition where the Reporting UAS will not include its URI is if it wants to keep its URI private. In this case the Reporting UAS MUST NOT include this tag in the IMDN.

10.5. original-recipient

If there is an Original-To header in the IM, the Reporting UAS MUST include the value of the Original-To header as the value to the original-recipient tag.

10.6. disposition-time

The Reporting UAS MUST include the disposition time reflecting when the reported disposition occured as the value of the disposition-time tag. The format of the time value MUST follow the format specified in RFC 3339 [10], using UTC.

11. Examples

11.1. Simple End-to-End IMDN Request

Request

```
From: Eric Burger <im:eburger@example.com>
To: Hisham Khartabil <im:hisham.khartabil@example.net>
DateTime: 2005-10-18T09:27:22-5
Subject: Did you get this?
NS: imdn <urn:ietf:params:cpim-headers:imdn>
imdn.Disposition-Notification:
imdn.Message-ID: 1542af3e8b@eburger@example.com

Content-type: text/xml; charset=utf-8
Content-ID: <1542af3e8b-12@eburger@example.com>

<body>
Did you get this message?
</body>
```

Response

```
From: Hisham Khartabil <im:hisham.khartabil@example.net>
To: <im:eburger@example.com>
DateTime: 2005-10-18T09:30:18+1
Subject: Did you get this?
NS: imdn <urn:ietf:params:cpim-headers:imdn>
imdn.Message-ID: latida27@stuff@example.net

Content-type: multipart/signed; boundary=next;
              micalg=sha1;
              protocol=application/pkcs7-signature

--next
Content-type: application/imdn+xml; charset=utf-8

<imdn>
  <disposition>read</disposition>
  <reporting-uas-uri>
    im:hisham.khartabil@example.net
  </reporting-uas-uri>
  <original-recipient-uri>
    im:hisham.khartabil@example.net
  </original-recipient-uri>
  <original-message-id>
    1542af3e8b@eburger@example.com
  </original-message-id>
</imdn>
--next
Content-type: application/pkcs7-signature

{signature stuff}
  :
  :
--next--
```

Note the IMDN plaintext would not have the CRLF's in the data elements. We do that here simply for readability.

11.2. Gateway Endpoint

Happy Path for gateway reporting it forwarded. Same request as above, but with processed response.

11.3. List Exploder - Forward

Happy Path for forwarding case. Note the different responses, but with same Original-To and Original-Message-Id.

11.4. List Exploder - Private Forward

Show no Original-Sender

11.5. List With Lists

Show wrapped, wrapped responses.

11.6. End-to-End Encryption Forwarded

Gateway scenario where Reporting UAS encrypts IMDN document for read only by Requesting UAC.

12. Formal Syntax

12.1. IMDN CPIM Request

TODO: collect syntax from above.

12.2. IMDN Document

Coming soon.

13. IANA Considerations

URN name in IETF namespace: urn:ietf:params:cpim-headers:imdn

IMDN schema in Section 12.2.

14. References

14.1. Normative References

[1] Klyne, G. and D. Atkins, "Common Presence and Instant Messaging (CPIM): Message Format", RFC 3862, August 2004.

[2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[3] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

[4] Campbell, B., "The Message Session Relay Protocol", draft-ietf-simple-message-sessions-13 (work in progress), December 2005.

[5] Fajman, R., "An Extensible Message Format for Message Disposition Notifications", RFC 2298, March 1998.

[6] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Certificate Handling", RFC 3850, July 2004.

[7] Yergeau, F., Paoli, J., Sperberg-McQueen, C., Bray, T., and E. Maler, "Extensible Markup Language (XML) 1.0 (Third Edition)", W3C REC REC-xml-20040204, February 2004.

[8] Moats, R., "URN Syntax", RFC 2141, May 1997.

[9] Moats, R., "A URN Namespace for IETF Documents", RFC 2648, August 1999.

[10] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.

[11] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

14.2. Informative References

[12] Klensin, J., "Simple Mail Transfer Protocol", RFC 2821, April 2001.

[13] Jennings, C., "Relay Extensions for the Message Sessions Relay Protocol (MSRP)", draft-ietf-simple-msrp-relays-06 (work in progress), December 2005.

[14] Hansen, T. and G. Vaudreuil, "Message Disposition Notification", RFC 3798, May 2004.

[15] Hansen, T., "Message Tracking Model and Requirements", RFC 3888, September 2004.

[16] Garcia-Martin, M. and G. Camarillo, "Multiple-Recipient MESSAGE Requests in the Session Initiation Protocol (SIP)", draft-ietf-sipping-uri-list-message-06 (work in progress), January 2006.

Appendix A. Contributors

Appendix B. Acknowledgements

Thanks go to Ben Campbell for continuously prodding me. Thanks also to Hisham for the relay idea and threatening some text to force me back to the task. Dean kept reminding me that 3GPP really, really wants this done and to work.

Author's Address

Eric Burger
Brooktrout Technology, Inc.
18 Keewaydin Dr.
Salem, NH 03079-2839
USA

Phone: +1 603 890 7587
Fax: +1 603 457 5944
Email: eburger@brooktrout.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.